FisherPoser: Human Motion Estimation from Sparse Observations with Hierarchical Region-Wise Fisher-Matrix Uncertainty Modeling

Songpengcheng Xia [1] Qingyu Zhang [1] Zhuo Su [2] Jiarui Yang[1] Zengyuan Lai [1]
Qi Wu[1] Ling Pei[1†] *
[1] Shanghai Jiao Tong University [2] ByteDance

This supplementary document provides additional details to support the main paper. Section A elaborates on the network architecture and training configuration. Section B describes auxiliary training losses for enhancing global consistency and motion smoothness. Section C provides a detailed introduction to Matrix–Fisher rotation modeling on $\mathrm{SO}(3)$, including its parameterization and the associated learning objective. Section D summarizes the datasets, evaluation protocols, and baseline implementations. Section E presents extended ablation studies and visualizations, including error distribution analysis and results on real-world VR sequences. Finally, Section F discusses limitations and potential future work.

A. Implementation Details

Input Representation. Each frame at time t receives the VR observation $\mathbf{x}_t \in \mathbb{R}^{45}$ and a SMPL-style pose history $\mathbf{p}_t \in \mathbb{R}^{198}$ (22 joints, each with a 3×3 rotation matrix flattened to 9D). The VR input encodes head and hand positions, linear velocities, and rotations.

Both $\mathbf{x}_t$ and $\mathbf{p}_t$ are embedded into a shared latent space of dimension 256 via two linear layers. The embeddings are summed and enriched with sinusoidal positional encodings. The resulting sequence is passed through a 3-layer Transformer encoder with 4 attention heads, feed-forward size 1024, GELU activation, and causal masking (so that each frame only attends to past frames). The final output hidden states $\{z_t\}$ are used as global context for subsequent modules.

Region-wise Representation and Semantic Anchors. We partition the 22 SMPL joints into five regions: torso, left/right arms, and left/right legs. A fixed joint-to-region map guides the assignment. Joint rotations are reshaped to $[B, T, 22, 9]$ and projected into 64D joint features via a linear layer.

Parallel to this, region-specific semantic anchors are constructed directly from the VR input. For torso and arms, we concatenate head position/velocity with head/hand rotations (arm rotations are relative to the head). For legs (which lack direct sensor data), we use weak priors such as head forward direction, head height, and speed. These raw features (15D for torso/arms, 5D for legs) are mapped into a 32D anchor space via small linear layers. Each region also has a learnable 16D embedding.

For each region r, we gather joint features as keys/values and form a query by concatenating z_t, the region anchor, and the region embedding. A single-head attention pooling module produces a pooled 128D feature per region and frame. This feature is passed through a 2-layer MLP to obtain a 256D region token $T_t^{(r)}$. Stacking these tokens results in $T_t \in \mathbb{R}^{5\times256}$.

Region-wise Matrix–Fisher Prediction. With the global context z_t and region tokens $T_t^{(r)}$, region-specific MLP heads are used to regress Matrix–Fisher parameters for all joints within each region. The concatenation of z_t and $T_t^{(r)}$ is passed through two 2-layer MLPs that output (i) an unconstrained 3×3 matrix per joint (flattened to 9D) and (ii) a 3D concentration logit per joint. These outputs are reshaped to give per-joint matrices $F_{\text{dir},t}^{(j)} \in \mathbb{R}^{3\times3}$ and raw concentrations $u_{\text{dir},t}^{(j)} \in \mathbb{R}^3$. This provides a region-aware yet per-joint independent Matrix–Fisher prediction.

Hierarchical Limb Refinement. To enforce kinematic dependencies, we refine predictions along the four limb chains (left/right arms and legs). Starting from the region-wise parameters $(F_{\text{pred},t}^{(p)}, u_{\text{pred},t}^{(p)})$ of the parent joint p, we construct a refinement feature for its child c by concatenating: (i) the global context z_t, (ii) the region token for c's region, (iii) the flattened parent matrix $F_{\text{pred},t}^{(p)}$, and (iv) the parent concentration $u_{\text{pred},t}^{(p)}$.

This feature is passed through child-specific MLPs to predict $(F_{\text{prop},t}^{(c)}, u_{\text{prop},t}^{(c)})$. The final predictions are a mixture

*Corresponding authors
This work was supported by the National Nature Science Foundation of China (NSFC) under Grant 62273229.

of direct and proposed predictions, using a fixed mixing coefficient $\lambda = 0.6$:

$$F^{(c)}\text{pred}, t = (1-\lambda)F^{(c)}_{\text{dir},t} + \lambda F^{(c)}_{\text{prop},t}, \quad (1)$$

$$u^{(c)}\text{pred}, t = (1-\lambda)u^{(c)}_{\text{dir},t} + \lambda u^{(c)}_{\text{prop},t}. \quad (2)$$

This refinement is applied once per frame over all limbs, with torso joints retaining their region-wise predictions.

Final Parameterization and Core Loss. After limb refinement, we reshape all $F^{(j)}_{\text{pred},t}$ back to 3×3 and exponentiate $u^{(j)}_{\text{pred},t}$ to obtain nonnegative axis-wise concentrations. These form the final Matrix–Fisher parameters used for training.

B. Additional Training Losses

Besides the Matrix–Fisher negative log-likelihood L_{MF} and the geodesic mode-alignment loss L_{mode} described in the main paper, we employ several auxiliary objectives that improve global consistency and temporal smoothness. All losses are applied to global SMPL joints after forward kinematics and are combined with tuned scalar weights.

Hand alignment loss. Following common practice, we derive global translation from the HMD by aligning the predicted head position to the measured HMD position to obtain a global full-body pose $\mathbf{p}_t$ at frame t. This alignment can introduce a mismatch between the predicted global hand locations and the controller measurements. Instead of using an additional IK module, we enforce consistency via a simple hand alignment loss:

$$L_{\text{hand}} = \frac{1}{2T}\sum_{t=1}^{T}\left(\left\|\mathbf{p}_t^{\text{L-h}} - \hat{\mathbf{p}}_t^{\text{L-h}}\right\|_1 + \left\|\mathbf{p}_t^{\text{R-h}} - \hat{\mathbf{p}}_t^{\text{R-h}}\right\|_1\right), \quad (3)$$

where $\mathbf{p}_t^{\text{L/R-h}}$ are the predicted global hand joint positions after head alignment, and $\hat{\mathbf{p}}_t^{\text{L/R-h}}$ are the corresponding ground-truth (controller) positions. This term forces the avatar's hands to remain tightly aligned with the input VR signals while keeping the whole framework end-to-end differentiable.

Motion smoothness and velocity matching. To encourage temporally coherent motion, we adopt a velocity-matching loss between frames at multiple temporal strides. For a stride $l \in \{1, 3, 5\}$, we define

$$L_v(l) = \frac{1}{T-l}\sum_{t=1}^{T-l}\left\|(\mathbf{p}_{t+l} - \mathbf{p}_t) - (\hat{\mathbf{p}}_{t+l} - \hat{\mathbf{p}}_t)\right\|_1, \quad (4)$$

where $\mathbf{p}_t$ and $\hat{\mathbf{p}}_t$ denote the predicted and ground-truth global joint positions, respectively. Using multiple strides $l = 1, 3, 5$ reduces both short-term jitter and longer-term drift, and avoids accumulating velocity errors when only l=1 is used.

Foot-contact loss. We further impose a foot-contact loss to suppress sliding when the feet are on the ground. Let $\mathbf{p}_t^{\text{feet}}$ be the subset of joints corresponding to the feet (ankles and toes), and $m_t \in \{0, 1\}^k$ be a binary contact mask indicating which foot joints are in contact at frame t. The foot-contact loss is

$$L_{\text{fc}} = \frac{1}{T-1}\sum_{t=1}^{T-1}\left\|(\mathbf{p}_{t+1}^{\text{feet}} - \mathbf{p}_t^{\text{feet}}) \odot m_t\right\|_1, \quad (5)$$

which penalizes non-zero foot velocities during contact frames and effectively reduces foot-sliding artifacts.

Overall objective. The full training objective is a weighted sum of the Matrix–Fisher NLL, geodesic mode loss, hand alignment loss, multi-stride velocity losses, and foot-contact loss:

$$\begin{aligned} L_{\text{total}} = {} & \lambda_1 L_{\text{MF}} + \lambda_2 L_{\text{mode}} + \lambda_3 L_{\text{hand}} \\ & + \sum_{l\in\{1,3,5\}} \lambda_4 L_v(l) + \lambda_5 L_{\text{fc}}. \end{aligned} \quad (6)$$

The hyper-parameters $\{\lambda_i\}_{i=1,\dots,5}$ are set to $\{1.0, 2.0, 1.0, 5.0, 2.0\}$. All coefficients $\lambda_\cdot$ are selected on a validation split and kept fixed across all experiments and ablations.

Optimization. All models are implemented in PyTorch and trained with AdamW. Unless otherwise stated, we use a batch size of 128 sequences, an initial learning rate of 3×10^{-4} with cosine decay and a 5-epoch linear warm-up, weight decay 10^{-4}, and gradient-norm clipping at 1.0. The Transformer encoder uses dropout rate 0.1 on attention and feed-forward layers. During training, input clips of length $T = 40$ frames are sampled with random starting indices from each sequence. For inference, full sequences are processed in a streaming fashion: we feed the sequence once from $t = 1$ to T, with the pose history at time t taken from the model's prediction at time $t - 1$.

Hardware and runtime. All experiments are conducted on a single workstation equipped with an NVIDIA RTX 4090 GPU and 24 GB of memory. FisherPoser and all baselines are implemented in PyTorch and trained with identical batch sizes, sequence lengths, and data augmentation strategies within each protocol. At inference time, FisherPoser runs in real time for 30 fps input sequences on the RTX 4090, and its computational cost is comparable to other transformer-based baselines.

C. Matrix–Fisher Distribution on SO(3)

This section provides background on the Matrix–Fisher distribution used in our model.

Definition and Parameterization. The Matrix–Fisher distribution defines a probability distribution over the special orthogonal group $\mathrm{SO}(3)$, with the density function:

$$p(R \mid F) = c(F)\, \exp\big(\mathrm{tr}(F^\top R)\big), \tag{7}$$

where $R \in \mathrm{SO}(3)$ is the rotation matrix and F is the parameter matrix. The normalizing constant $c(F)$ ensures that the integral over $\mathrm{SO}(3)$ is 1. Using SVD, F can be decomposed as $F = U\,\mathrm{diag}(s)\,V^\top$, where $U, V \in \mathrm{SO}(3)$ and s are the singular values. This decomposition allows us to express the distribution's mode and axis-wise concentration, which are key to modeling the uncertainty of rotations.

Mode and Concentration. The mode of the Matrix–Fisher distribution is given by $R^\star = UV^\top$. The singular values s_1, s_2, s_3 determine the concentration of the distribution, with larger values indicating higher confidence in the rotation along that axis. Smaller values correspond to greater uncertainty, particularly for the less constrained axes.

Negative Log-Likelihood and Approximation. The Matrix–Fisher negative log-likelihood is given by:

$$L_{\text{MF-NLL}}(R_{\text{gt}}, F) = -\mathrm{tr}(F^\top R_{\text{gt}}) + \log c(F), \tag{8}$$

where R_{gt} is the ground-truth rotation. The normalizing constant $c(F)$ is approximated using an eigenvalue-based series expansion, which is differentiable and efficiently implemented across all joints and time steps in our network.

Relation to Gaussian Models. The Matrix–Fisher distribution can be approximated by a Gaussian around its mode in the Lie algebra $\mathrm{So}(3)$, where the covariance is inversely proportional to the concentration parameters. This makes the Matrix–Fisher distribution a suitable model for uncertainty-aware rotation estimation, as it provides a continuous representation of rotations on $\mathrm{SO}(3)$ with geometrically meaningful concentration parameters.

D. Datasets, Metrics, and Baselines

Datasets and protocols. All experiments are conducted on AMASS [11], a large-scale collection of motion-capture sequences retargeted to the SMPL body model. We follow two widely adopted protocols from recent HMD-based avatar works [2, 3, 5, 8, 20].

Protocol 1 (P1). We follow the setting used in [5, 7, 8, 20] and focus on three AMASS subsets: CMU [18], BMLrub [15], and HDM05 [13]. Each subset is randomly split into 90% training and 10% testing sequences. We then synthesize head and hand tracker signals from the SMPL body as in prior work and train/evaluate all methods on this split.

Protocol 2 (P2). Following the larger-scale benchmark in [3, 5], we aggregate multiple AMASS components for training, including ACCAD [1], CMU [18], MoVi [6], MPI [9], EYES [10], KIT [12], HDM05 [13], TCD [16], TotalCapture [17], and SFU [19]. The Transitions [11] and HumanEva [14] subsets are held out for evaluation only. Compared to P1, this protocol exposes the model to a broader distribution of motions and subjects.

For both protocols, sequences are resampled to 60 fps. During training we sample overlapping clips of length T=40 frames with random starting indices; at test time we run on full sequences in a streaming fashion. Head and controller 6-DoF signals are derived from SMPL joints and used as the only inputs for all methods, ensuring strictly comparable sparse VR observations.

Evaluation metrics. We report four main metrics that jointly measure spatial accuracy and temporal quality.

MPJRE (Mean Per-Joint Rotation Error, °) clculates the average differences between predicted joint rotations and the ground-truth rotations, which reports the pose accuracy of motion estimation.

MPJPE (Mean Per-Joint Position Error, mm) is the Euclidean distance between predicted and ground-truth 3D joint positions in the global frame, averaged over all joints and time steps:

$$\text{MPJPE} = \frac{1}{TJ} \sum_{t,j} \left\| \mathbf{p}^{(j)}_{t,\text{pred}} - \mathbf{p}^{(j)}_{t,\text{gt}} \right\|_2. \tag{9}$$

MPJVE (Mean Per-Joint Velocity Error) measures the discrepancy between predicted and ground-truth joint velocities, and reflects temporal smoothness at the first-order level. We compute finite differences $\mathbf{v}^{(j)}_t = \mathbf{p}^{(j)}_{t+1} - \mathbf{p}^{(j)}_t$ and report the mean ℓ_2 error between predicted and ground-truth velocities across joints and frames.

Jitter is a higher-order smoothness metric adapted from [2, 3]. We approximate joint accelerations and jerks using second- and third-order finite differences of 3D joint positions, and define jitter as the mean ℓ_2 norm of the jerk across all joints and frames. This metric is particularly sensitive to high-frequency artifacts (e.g., frame-to-frame flips or micro-oscillations) and correlates well with perceptual motion stability.

For all metrics, lower values are better.

Competing methods. We compare FisherPoser with a range of state-of-the-art full-body pose estimators from

sparse VR signals, re-implementing or adapting public code where necessary so that all methods operate on the same HMD + controllers input:

AvatarPoser [8] is a deterministic transformer-based regressor that maps sparse VR tracking signals to SMPL pose. It uses a temporal transformer encoder over head and hand trajectories and directly regresses a single full-body pose per frame.

AGRoL [4] (Avatars Grow Legs) is a diffusion-based generative model that synthesizes long-horizon full-body motion from sparse tracking. It models a denoising process in pose space to produce temporally smooth sequences and can sample multiple plausible motions for the same input.

AvatarJLM [20] introduces a joint-level modeling framework that decouples global pose and local joint details. It employs a two-stage architecture where an initial pose is regressed from sparse signals and then refined using joint-wise transformers, leading to realistic and temporally coherent avatars.

SAGE [5] (Stratified Avatar Generation from Sparse Observations) proposes a stratified generative architecture that separately handles upper- and lower-body factors. It leverages diffusion-based generation with learned stratification to improve lower-body plausibility under sparse head–hand observations.

HMDPoser [3] is a strong VR baseline that focuses on accurate and efficient full-body motion estimation from a minimal HMD + controller setup. It combines a temporal network with carefully designed priors and contact-aware objectives to improve lower-limb reconstruction and motion stability.

RPM [2] (we follow the authors' rolling-prediction implementation) is a rotation-parameterized model designed for long-horizon full-body motion from temporally and spatially sparse inputs. It emphasizes stable rotation modeling and temporal consistency, yielding very low jitter and velocity errors, sometimes at the cost of reduced pose accuracy.

All baselines are trained under our P1 and P2 protocols using their recommended hyperparameter settings whenever possible. For methods that support additional trackers (e.g., pelvis or feet), we disable those inputs and retrain them with only three 6-DoF VR trackers (HMD and two controllers) to match our setting.

E. Additional Analysis and Qualitative Results

E.1. Extended Ablation Discussion

Component-wise ablation. Tab. 2 in the main paper reports ablations over four variants: the deterministic autoregressive backbone (Ours-AR), the same backbone with a Matrix–Fisher head (Ours-Fisher), the addition of region-wise conditioning (Ours-Fisher-Part), and the full model with limb-level hierarchical refinement (Ours). Moving from Ours-AR to Ours-Fisher already reduces rotation and position errors and yields smoother motion, highlighting the benefit of calibrated, geometry-aware uncertainty. Introducing region tokens (Ours-Fisher-Part) produces the largest gain in both accuracy and jitter by aligning model capacity with heterogeneous observability across torso, arms, and legs. Finally, the hierarchical variant (Ours) further improves all metrics, indicating that probabilistic modeling, regional conditioning, and parent-to-child propagation are complementary rather than redundant.

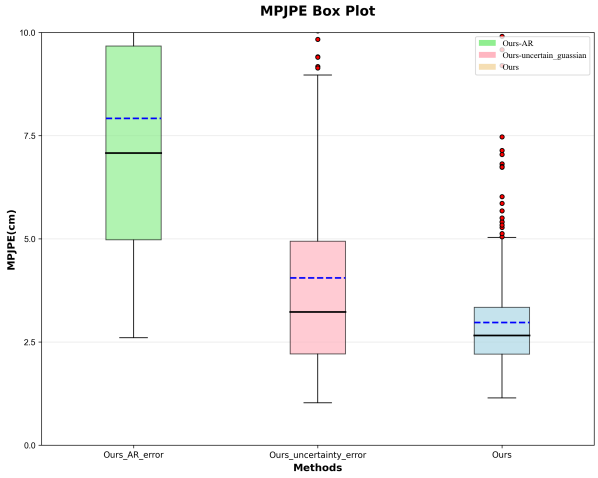

Figure 1. MPJPE error distribution (cm) for the deterministic baseline (Ours-AR), Gaussian-uncertainty variant, and our full FisherPoser. Black: mean; blue dashed: median.

Uncertainty parameterization. Tab. 3 in the main paper compares different uncertainty heads while keeping the causal Transformer fixed: a deterministic regressor (Ours-AR), a heteroscedastic Gaussian head in axis–angle space (Ours-AR-Gaussian), and our Matrix–Fisher head on $\mathrm{SO}(3)$ with region tokens and limb refinement (Ours). As shown in the table, introducing uncertainty helps: the Gaussian NLL head yields lower rotation and position errors than the deterministic baseline. However, it noticeably harms temporal stability, with higher MPJVE and jitter, indicating a mismatch between Euclidean parameterization and rotation geometry. Our Matrix–Fisher variant overcomes this trade-off: it achieves higher accuracy while restoring smooth motion by modeling rotations on $\mathrm{SO}(3)$ and separating the mode from axis-wise concentration. Moreover, the comparison with Ours-Fisher in Tab. 2, which only adds a Matrix–Fisher output head, further confirms that manifold-aware uncertainty yields better-calibrated predictions.

Moreover, Fig. 1 shows MPJPE error distributions for three variants: Ours-AR, Ours-uncertainty-Gaussian, and our full FisherPoser. In each box, the black line denotes

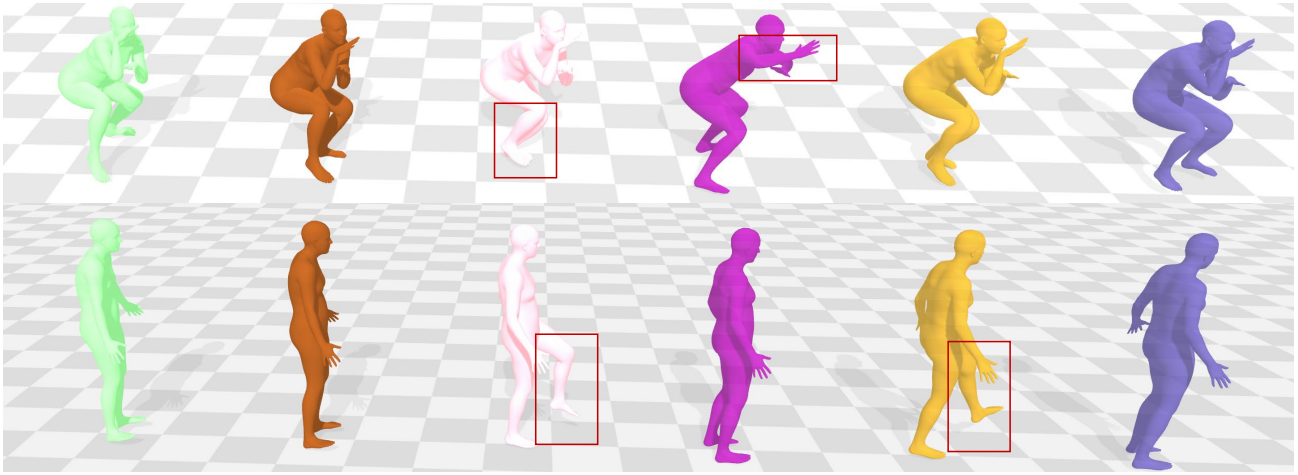

Figure 2. Qualitative comparison on two held-out test sequences from the CMU dataset. We show ground truth, our FisherPoser, and four strong baselines. Our method yields more accurate lower-limb articulation and foot contacts, with fewer leg-collapse and foot-floating artifacts.

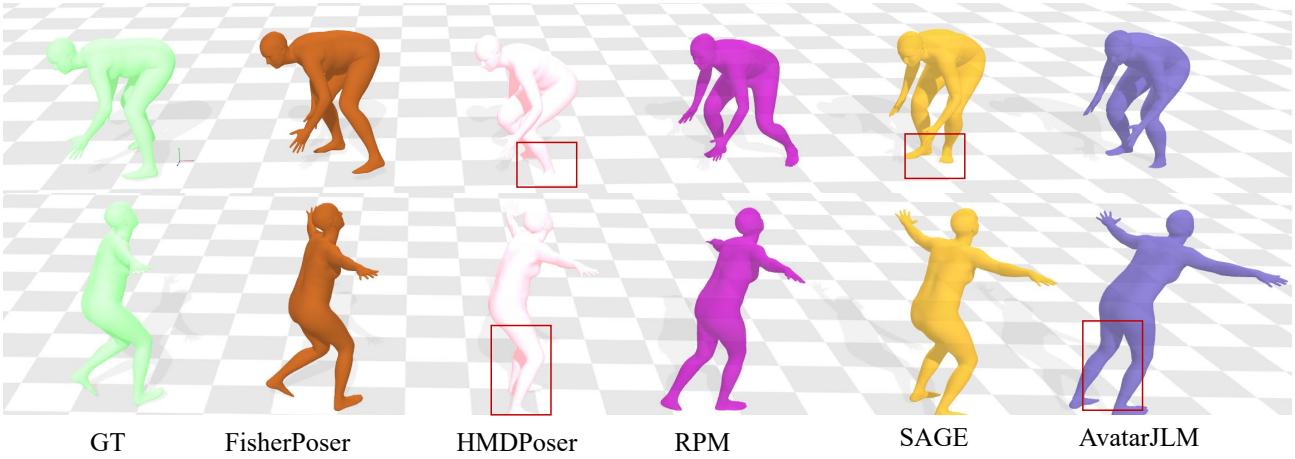

Figure 3. Additional qualitative comparison on two other CMU sequences, covering walking, turning, reaching, and high-dynamic motions. FisherPoser produces smoother temporal evolution and more realistic contact behavior than competing approaches.

the mean error and the blue dashed line marks the median. The purely deterministic baseline (Ours-AR) exhibits both the largest mean/median error and a very wide interquartile range with many high-error outliers, indicating frequent failure cases. Introducing a Gaussian uncertainty head (Ours-uncertainty-Gaussian) shifts both the mean and median downward and visibly compresses the spread of errors, suggesting that even a simple uncertainty model helps reduce large mistakes. Our full model achieves the lowest mean and median MPJPE, with the tightest box and the fewest extreme outliers, demonstrating not only improved overall accuracy but also a more concentrated error distribution and fewer catastrophic frames.

E.2. Additional Qualitative Comparisons

More Visualization. Fig. 2 and Fig. 3 provide further qualitative comparisons on test sequences from the CMU dataset. For each sequence we visualize the ground truth, our FisherPoser, and four strong baselines (HMDPoser, AvatarJLM, SAGE, and RPM) over multiple frames. The examples cover a wide range of motions, including deep squats, walking and turning, reaching, and high-dynamic running and jumping. Consistent with the main paper, our method produces more accurate lower-limb articulation and foot contacts, avoids the foot-floating and leg-collapse artifacts frequently observed in competing methods, and maintains smoother temporal evolution of the pose. These visualizations further corroborate that FisherPoser yields both more realistic body configurations and more stable motion over time.

Real-world VR data. We further deploy our model on sequences captured with consumer VR hardware. For each sequence, we record only the three 6-DoF trajectories from

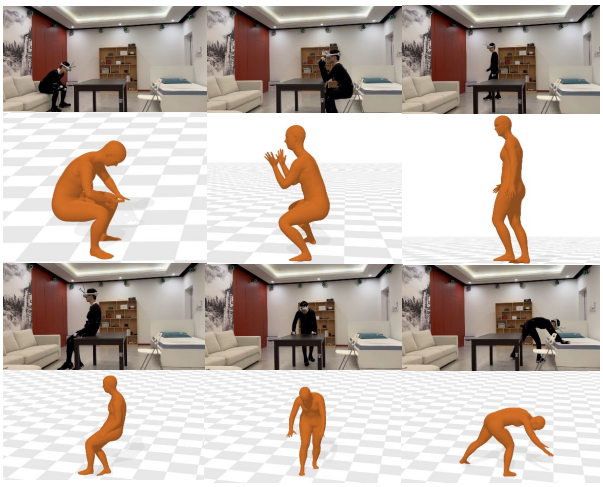

Figure 4. Qualitative results on sequences captured with consumer VR devices. From only three 6-DoF trackers, FisherPoser reconstructs plausible, temporally coherent full-body motion in real-world settings.

the HMD and two hand controllers and feed them directly to our network to reconstruct full-body motion. Fig. 4 shows representative frames of the recovered avatars under various in-the-wild motions. The results indicate that our method produces plausible and temporally coherent full-body poses from real VR signals without any dataset-specific fine-tuning; additional examples are provided in the supplementary video.

F. Discussion and Limitations

FisherPoser demonstrates that geometry-aware uncertainty modeling and region-wise hierarchical decoding can substantially improve sparse VR-based full-body motion capture. Nonetheless, several limitations remain, which also suggest avenues for future work:

- **Runtime and deployment constraints.** The current architecture is lightweight enough for real-time inference on a modern GPU, but direct deployment on standalone VR headsets with limited compute and power budgets remains non-trivial. Model compression, distillation, and architecture co-design for on-device execution are important next steps if FisherPoser is to be used in consumer-facing applications.
- **Sensing configuration and environment interaction.** Our framework assumes the minimal three-tracker setting (HMD and two controllers) and does not explicitly model the static environment or dynamic human–object interactions. In many practical scenarios, however, contact with the floor, walls, furniture, or handheld objects provides strong constraints on otherwise unobserved joints. Integrating richer anchors—such as terrain and contact cues, scene meshes, or object poses—into the region-wise conditioning and hierarchical decoding is an important extension, especially for complex interactive behaviors.
- **Uncertainty usage and multi-hypothesis prediction.** In this work, Matrix–Fisher distributions are primarily used to obtain calibrated point estimates and confidence measures. A natural next step is to exploit these distributions more fully, for example by sampling multiple pose hypotheses or combining them with diffusion-style refinement, particularly in highly ambiguous configurations or during human–object interactions. This would align our probabilistic formulation more closely with downstream tasks such as interaction prediction, planning, or safety-aware control.

References

[1] Ijaz Akhter and Michael J Black. Pose-conditioned joint angle limits for 3d human pose reconstruction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1446–1455, 2015. 3

[2] German Barquero, Nadine Bertsch, Manojkumar Marramreddy, Carlos Chacón, Filippo Arcadu, Ferran Rigual, Nicky Sijia He, Cristina Palmero, Sergio Escalera, Yuting Ye, et al. From sparse signal to smooth motion: Real-time motion generation with rolling prediction models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1850–1860, 2025. 3, 4

[3] Peng Dai, Yang Zhang, Tao Liu, Zhen Fan, Tianyuan Du, Zhuo Su, Xiaozheng Zheng, and Zeming Li. Hmd-poser: On-device real-time human motion tracking from scalable sparse observations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 874–884, 2024. 3, 4

[4] Yuming Du, Robin Kips, Albert Pumarola, Sebastian Starke, Ali Thabet, and Artsiom Sanakoyeu. Avatars grow legs: Generating smooth human motion from sparse tracking inputs with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 481–490, 2023. 4

[5] Han Feng, Wenchao Ma, Quankai Gao, Xianwei Zheng, Nan Xue, and Huijuan Xu. Stratified avatar generation from sparse observations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 153–163, 2024. 3, 4

[6] Saeed Ghorbani, K Mahdaviani, Anne Thaler, K Kording, DJ Cook, G Blohm, and NF Troje. Movi: A large multipurpose motion and video dataset. arxiv 2020. *CoRR*, 9, 2020. 3

[7] Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. Motiongpt: Human motion as a foreign language. *Advances in Neural Information Processing Systems*, 36:20067–20079, 2023. 3

[8] Jiaxi Jiang, Paul Streli, Huajian Qiu, Andreas Fender, Larissa Laich, Patrick Snape, and Christian Holz. Avatarposer: Articulated full-body pose tracking from sparse motion sensing. In *European conference on computer vision*, pages 443–460. Springer, 2022. 3, 4

[9] Matthew Loper, Naureen Mahmood, and Michael J Black. Mosh: motion and shape capture from sparse markers. *ACM Trans. Graph.*, 33(6):220–1, 2014. 3

[10] Eyes JAPAN Co. Ltd. Eyes japan mocap dataset. 3

[11] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5442–5451, 2019. 3

[12] Christian Mandery, Ömer Terlemez, Martin Do, Nikolaus Vahrenkamp, and Tamim Asfour. The kit whole-body human motion database. In *2015 International Conference on Advanced Robotics (ICAR)*, pages 329–336. IEEE, 2015. 3

[13] Meinard Müller, Tido Röder, Michael Clausen, Bernhard Eberhardt, Björn Krüger, and Andreas Weber. Mocap database hdm05. *Institut für Informatik II, Universität Bonn*, 2(7), 2007. 3

[14] Leonid Sigal, Alexandru O Balan, and Michael J Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International journal of computer vision*, 87(1):4–27, 2010. 3

[15] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. Human motion diffusion model. *International Conference on Learning Representations (ICLR)*, 2023. 3

[16] Nikolaus F Troje. Decomposing biological motion: A framework for analysis and synthesis of human gait patterns. *Journal of vision*, 2(5):2–2, 2002. 3

[17] Matthew Trumble, Andrew Gilbert, Charles Malleson, Adrian Hilton, and John Collomosse. Total capture: 3d human pose estimation fusing video and inertial sensors. In *Proceedings of 28th British Machine Vision Conference*, pages 1–13, 2017. 3

[18] Carnegie Mellon University. Cmu mocap dataset. 3

[19] Simon Fraser University and National University of Singapore. Sfu motion capture database. 3

[20] Xiaozheng Zheng, Zhuo Su, Chao Wen, Zhou Xue, and Xiaojie Jin. Realistic full-body tracking from sparse observations via joint-level modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14678–14688, 2023. 3, 4